Automatic Retrieval of Frequent Idiomatic and Collocational Expressions in a Large Corpus

Y. Choueka, S. T. Klein and E. Neuwitz

1. Introduction and Motivation

One of the interesting lists that a researcher would like to receive as a by-product of the automatic processing of a large corpus would certainly be a list of common or text-specific idioms, expressions, collocations and the like, that occur frequently enough in that corpus. By these terms we mean more specifically a sequence of two or more *consecutive* words that constitutes an autonomous linguistic unit and has acquired, because of recurrent use in specialized contexts, a meaning or a connotation that somehow transcend the ordinary meaning of its constituents. Such sequences can be of several different types. They can be well-established idioms, i.e. 'Succession of words whose meaning must be learnt as a whole' (*The Advanced Learner's Dictionary of Current English*, 2nd Edition), such as *once upon a time, by and large, time and again*; expressions that are not genuinely idiomatic but still are used more often than not as a unique syntactical and semantical unit, such as *research and development, Merry Christmas, hit and run*; compound names of entities, such as *United Nations, Security Council*; or foreign expressions such as *curriculum vitae, status quo*, and the like. We call such sequences, in short, *expressions*. A good rule of thumb to decide whether a sequence indeed qualifies as such an expression or not, is whether one can justify the introduction of a specific entry for that sequence in a comprehensive concordance of the corpus, rather than listing it under the heading of one of its constituents. Another adequate criterion can be whether a learned informant can guess (knowing that he is dealing with an expression) the entire sequence once he has read (or heard) its beginning. The examples above would probably qualify as 'expressions' under both criteria.

2. Preliminary Considerations

In looking for an algorithm that would automatically produce a list of some selected expressions of that kind from a given corpus, we assume (as is usually indeed the case) that no operational high-quality automatic syntactical or semantical component is available in the system. The algorithms will have therefore to be based, necessarily, on the statistical aspects and the combinatorial properties of the words' distributions in the text. In consequence we shall have to restrict our approach to quite large corpora of a few hundred thousands, or even millions, of words. Also, only expressions that occur frequently enough in the corpus will stand any chance of being recognized and retrieved. (For a survey on algorithms for retrieving co-occurrences and collocations in a somewhat different sense than meant here, see [4] and the references cited there.)

Although motivated by a few general and intuitively plausible linguistic considerations, our approach was essentially experimental. Every suggested algorithm or formula was tested on a large corpus, the produced lists were examined, and appropriate changes were introduced in the algorithm. All the variations of the basic formula to be described below were tested on the database of the RESPONSA Project run at Bar-Ilan University (Israel). The RESPONSA database consisted then (spring 1981) of the full and unaltered text of 176 volumes of Rabbinical documents, each document being in fact a juridicial decision given according to the Jewish-Talmudic legal system, and related to an actual case presented to a Rabbinical court or brought to the attention of a prominent Rabbi. The database, spanning more than a thousand years and originating from more than twenty countries, contained then 37,500 documents, about 28 million words of running text. The language of the database is Hebrew, although the text contains quite a few Talmudic-Aramaic phrases and grammatical structures, and occasionelly also some vernaculars (for more information about the RESPONSA project and its database, see [1] and [2]).

Early in our research we decided to restrict ourselves to the retrieval of expressions (or initial parts of expressions), of length two. Besides the obvious computational savings that are allowed by this restriction (working with expressions of length 3 or more would have required exorbitant computer's time and memory resources), we assumed that most of the longer expressions would be identifiable anyway by their beginnings (as in: *once upon ...*; recall the second criterion mentioned above) and, in any case, compiling a file of such two-words sequences would be an adequate basis for further processing if needed. Finally our text-corpus was, as mentioned above, in Hebrew, and Hebrew morphology is rich in structure and combinations. Thus, prepositions and combinations of prepositions such as *the, in, from, to, and, that, when, and that in*, etc. as well as verbal personal pronouns (*I, you, ..., they*) can be attached as prefixes to the corresponding nouns and verbs, while possessive pronouns (*mine, your,..., their*) or accusative ones (*me, you,..., them*) can be attached as suffixes. Phrases as *and like the heaven's stars* translate to a two-words sequence in Hebrew. Restricting the processing to two-words expressions does not much affect, therefore, the experiment's scope.

Our first approach to locate and retrieve frequent two-words expressions in the RESPONSA database was, naturally enough, to compile a list of the most frequent consecutive pairs of words in the text, sorted by decreased frequency of the pair. The result was however a total failure. Out of the 50 most frequent pairs in the corpus, none would really qualify as a genuine expression and only four pairs, at the very best, would somehow satisfy the first criterion, but certainly not

the second one. Table 1 gives an English translation of the twenty most frequent pairs, together with the pair's frequency and the pair constituents' ranks in the frequency list of the corpus' different words. One can see for example that even the three most common pairs with about 18,000 occurrences (!) do not represent any specific linguistic or semantical unit. This situation might seem somewhat strange at first sight but can be readily accounted for. As it turns out, the most frequent pairs are formed by 'accidental' concatenation, so to speak, of the most frequent words. To take just one example, the words 'not' and 'this' occur 450,000 and 220,000 times respectively in the RESPONSA corpus. No wonder then that they co-occur in a sequence ('this not') about 6000 times. Even if we were dealing with randomly distributed elements, the expectancy of such a coincidence to occur would be: (450,000*220,000)/38,000,000=2600. Obviously there are still some elements of morphological, or, more generally, linguistic aspects in these combinations; the converse peir 'not this', which statistically has the same probability of occurrence as the original one, occurs only 220 times. Still the most frequent pairs are more conditioned by the high frequency of their constituents than by anything else. This is clearly indicated by the fact that from the 100 words that appear in the 50 most common pairs, 82 are themselves from the list of the 50 mos' common words, 8 more have frequency's ranks between 51 and 100, and only 10 have a rank of more than 100. To put it differently, of the 50 most common pairs, all but one pair, i.e. 98% of the pairs, have at least one component in the 1–50 ranks, and 66% have *both* components in the 1–50 ranks. This is also transparent from the pertial list given in Table 1.

It should be noted that qualitatively similar results were received when processing a French text ('*Journal de Jeunesse*' by L. Groulx; 215,000 words). A list of some of the very frequent pairs did not reveal any specially interesting combinations (see [3] for more details).

A different approach should therefore be developed.

3. Development of an Adequate Formula

The starting point of our approach was the following simple observation. Suppose a word w_1 occurs 50 times in a given corpus, and is invariably followed in all its occurrences there by the *same* word w_2. Undoubtedly we have a potentially interesting expression $w_1\,w_2$, even if w_2 is preceded in the totality of its occurrences in the corpus by a large number of different neighbours. We therefore shift our attention from the frequencies of pairs of words and of their constituents to what might be called the 'selectivity index' of a word w relatively to its following neighbour or to its preceding one. (We prefer this terminology to the somewhat more confusing one of right- and left- neighbours, which is dependent on the writing direction of the language being studied; what is 'left' for English or French is 'right' for Hebrew or Arabic and might be 'up' or 'bottom' for other languages). The idea is then to assign to every (different) word of the corpus a *neighbour-selectivity index* (NSI) that somehow correlates its frequency with the number of its *different* neighbours and the frequencies of the corresponding pairs. The words with the highest NSI's would presumably be the beginnings of interesting sequences, and together with their most frequent neighbours they would constitute potential candidates for expressions or initial parts of expressions.

From now on we restrict our attention to *following* neighbours only, called hereafter just neighbours. (We shall briefly refer to preceding neighbours at the end of the paper.)

Several variations of a basic formula for the NSI of a word were suggested, and the resulting lists examined and evaluated. According to the results received, more improvements were incorporated into the formula until the final version $s_3(w)$ given below was received and judged satisfactory. The adequacy of the formula was then objectively evaluated by a fully-documented test described together with its results in section 4 below.

The following requirements were chosen as guidelines for the definitions of the NSI function s(w) of a word w:

1. s(w) is a function of the frequency f(w) of w, of the number of its different neighbours d(w) and of the frequencies of these neighbours.
2. s(w) is an increasing function of 'neighbour-selectivity'. The more selective is w in regard to its neighbours, the larger will s(w) be, and the greater will be the plausibility of w being an expression's beginning.
3. As a function of probabilistic nature, we would like s(w) to range between 0 and 1. This is obviously no restriction since we are interested anyway in relative degrees of selectivity.
4. If all the neighbours of w are identical (d(w) = 1), then s(w) should be maximal, i.e. s(w) = 1. On the other hand, if all w-neighbours are different (d(w) = f(w)) then s(w) should be zero.
5. The conditions in 4 are contradictory if w occurs only once in the corpus; we therefore assume s(w) to be undefined if f(w) = 1, and anyway even if it is, we are not interested in its value for that case.

A simple arithmetic manipulation gives the following function as a first suggestion:

$$s_1(w) = \frac{f(w) - d(w)}{f(w) - 1}$$

All stipulated conditions are obviously satisfied. $0 \leqslant s_1(w) \leqslant 1$; $s_1(w) = 1$ if and only if $d(w) = 1$; $s_1(w) = 0$ if and only if $f(w) = d(w)$, and $s_1(w)$ is undefined when $f(w) = 1$.

Consider however a situation in which we have two words w_1 and w_2 both with frequency 100 and with 2 different neighbours, but with neighbours' frequencies 99 and 1 for w_1, and 50, 50 for w_2. In both cases $s_1(w) = 98/99 = .99$, although we would certainly like $s_1(w_1)$ to be much higher than $s_1(w_2)$. This is in accordance with the general approach outlined here and specially with the second criterion mentioned above that enables us to retrieve at most only one expression (if possible at all) that begins with a given w. Referring to the numerical example just given, one can say in a very simplistic way that the probability of

guessing w_2's most common neighbour is 0.99, as against 0.50 for w_1 neighbours. More importance should be attached therefore in s(w) to the local frequency m(w) of the most common neighbour of w ('local frequency' meaning its frequency as a neighbour of w). Instead of comparing f(w) to d(w) we would like therefore to compare m(w) to the local frequencies $n_i(w)$ $(1 \leqslant i \leqslant d(w)-1)$ of the other naighbours of w, or rather to the mean n(w) of these frequencies: $n(w) = \Sigma n_i(w)/(d(w)-1)$. (By definition, we put $n(w) = 0$ if $d(w) = 1$, i.e., if there are no 'other' neighbours.) We therefore now suggest:

$$s_2(w) = \frac{m(w) - n(w)}{f(w)}$$

Again $0 \leqslant s_2(w) \leqslant 1$; $s_2(w) = 1$ if and only if $m(w) - n(w) = f(w)$, that is (because $m(w) \leqslant f(w)$) if and only if $n(w) = 0$ and $m(w) = f(w)$, i.e. $d(w) = 1$; finally $s_2(w) = 0$ if and only if $m(w) = n(w)$ and in particular, if all the neighbours of w are different then $s_1(w) = 0$.

Consider now the case of two words w_1, w_2 each with three different neighbours with local frequencies 100, 99, 1 and 100, 50, 50 respectively. Taking into account the remark above that a given w will generally retrieve one expression at most, and our interpretation of s(w) as a guessing probability of some sort, we would expect $s_2(w_1) < s_2(w_2)$. However $s_2(w_1) = s_2(w_2) = 0.25$. This is so because the difference $m(w) - n(w)$ does not take into account the variation of the other neighbours' frequencies about the mean. We have therefore to divide $s_2(w)$ by some measure of the variation about the mean, and a natural choice for that is the standard deviation; $SD(w) = (1/(d(w)-1)^*\Sigma[n(w) - n_i(w)]^2)^{1/2}$. For normalization purposes we again divide the standard deviation by the mean to get the formula:

$$s_3(w) = \frac{m(w) - n(w)}{f(w)(1 + SD(w)/n(w))}$$

(whare 1 has been added as a correcting term to SD(w)/n(w) in case this term is zero, and to assure the proper range for $s_3(w)$, since SD(w) may be less than n(w)).

Again $0 \leqslant s_3(w) \leqslant 1$ (since the maximal value of $s_3(w)$ is when $m(w) = f(w)$ so that $n(w) = 0$ and $SD(w) = 0$); $s_3(w)$ is zero in the same conditions as before, and $s_3(w) = 1$ if and only if $m(w) = f(w)$, $n(w) = 0$.

This formula gives however too great a weight to large numbers of very rare neighbours that occur only once or twice (neighbours that can be in fact misspellings or entry errors). Thus two words w_1, w_2 with local neighbours' frequencies of 500, 100, 1, 1, 1, 1 and 500, 100, 1 respectively will have $s_3(w_1) = 0.2732$, $s_3(w_2) = 0.3777$ while we expect $s_3(w_1)$ to be very nearly equal to $s_3(w_2)$ (or only slightly smaller).

To minimize the influence of such neighbours it was decided to replace all neighbours with small local frequencies (less than some predetermined M_1) by one 'virtual' neighbour whose frequency is the mean of that of the small neighbours. Denoting the new number of neighbours, the new mean, the new frequency and the new standard deviation by d'(w), n'(w), f'(w) and SD'(w) we get $s_4(w)$ which is similar to $s_3(w)$ except for the primed letters.

Two problems remain with $s_4(w)$. First (and this is pertinent to $s_2(w)$ and $s_3(w)$ too) it gives value zero or near zero for every case in which there is a uniform or almost uniform distribution of local neighbours' frequencies; second, it does not take into account explicitly the number of different neighbours of w (two words with neighbours' local frequencies 100,5 and 100,5,5,5 respectively will get the same $s_4(w) = 0.9047$, if we take $M_1 = 6$).

It was decided therefore to take some linear combination of $s_1(w)$ and $s_4(w)$, and in the absence of any argument to the contrary we chose $\frac{1}{2},\frac{1}{2}$ as coefficients. In this way we take into account both the number of different neighbours (through s_1) and their distribution (through s_4). We therefore finally suggest:

$$s_5(w) = \frac{1}{2}\frac{f(w) - d(w)}{f(w) - 1} + \frac{1}{2}\frac{m(w) - n'(w)}{f(w)\left(1 + \dfrac{SD'(w)}{n'(w)}\right)}$$

where: f(w) is the number of occurrences of w, d(w) the number of its different neighbours, m(w) the frequence (as a neighbour) of its most common neighbour, n'(w) is the mean of the local neighbours' frequency (where all neighbours with local frequency less than M_1 are considered as a unique neighbour with local frequency equal to the corresponding mean), f'(w) and SD'(w) the corresponding corrected total frequency and standard deviation. If there is only one neighbour, the formula reduces to:

$$s_5(w) = \frac{1}{2}\frac{f(w) - 1}{f(w) - 1} + \frac{1}{2}\frac{f(w) - 0}{f(w)(1 + 0)} = 1.$$

On the other hand if all neighbours are different then:

$$s_5(w) = \frac{1}{2}\frac{f(w) - f(w)}{f(w) - 1} + \frac{1}{2}\frac{1 - 1}{f(w)(1 + 0)} = 0$$

4. Test

The last formula was tested on the RESPONSA database, whose relevant characteristics for our purposes are 38 million words of running text and about half-a-million different words. In order to make the computations practical we decided not to compute $s_5(w)$ for any w with less than 50 occurrences, or all of whose neighbours have less than 10 local occurrences. This elimination reduced the set of words to be handled to about 30,000 words. The parameter M_1 was moreover set to 10.

The values of $s_5(w)$ were computed for all pertinent w and a list was produced, giving the word w, its neighbour-selectivity-index, and its most common following neighbour, sorted by decreasing NSI. Even a cursory look on this list was sufficient to reveal that almost all of the few hundred pairs at the top of the list were indeed expressions or beginnings of expressions. Table 2 gives a sample of a few (non-consecutive) lines of the NSI list, together with the pertinent statistics (the corresponding words are not shown). Some interesting features of the behaviour of NSI are reflected in this table. We see for example that the beginning of the table consists of words with very few different neigh-

bours (1 to 3) and so have very high NSI (even though their frequencies are low: 100–300 occurrences). One notable exception is at line 7, where a word with 28 different neighbours appear with $s(w) = 0.9585$, preceding for example the word on line 10 with 2 different neighbours only. This is because all of the first word's neighbours are 'small' ones with frequency 1 or 2 while the second neighbour's of the second word has a local frequency of about 14% of the word's frequency. The full list, with the corresponding pairs of words, showed very convincingly the strong correlation between NSI values and expressions; the further we went down the list, the more we encountered non-expressions or accidental sequences.

Obviously such an intuitive assessment of the 'expression' quality of the pair and its correlation with the NSI (as defined by the function s_5) was not really satisfactory. We therefore devised a test which was supposed to measure in some more objective and quantitative terms the correlation mentioned above. The test was basically related to the second criterion described at this paper's beginning, namely whether the full expression can be guessed by its first word (s).

A list of the first 300 words grouped into six sets of 50 words each of the NSI list mentioned above (sorted by decreasing NSI), was printed, omitting this time the most common neighbour of each word. The list was handed to six different informants with considerable background in Responsa literature and Talmudic writings. They were asked to record besides each word in the list their guess as to its most common neighbour, assuming that they were dealing with expressions. The results are summarized in Table 3, where the percentage of correct guesses for the three best informants, and the corresponding average, are recorded. One can see that the percentage of correct guessing — for the three informants — is very high for the beginning of the list (87%) and it declines consistently with the decreasing NSI; for the portion of the list with NSI value of 0.60 to 0.63, the rate is about 50% only.

In order to further test the validity of our approach, three more groups of 50 words each were given to the informants. The first one consisted of words taken from the very end of the NSI-decreasing list, where NSI values were in the range of 0.13 to 0.17. The second contained 50 words chosen at random from the list; and the third consisted of the leading words of the 50 most frequent pairs in the whole database, with frequencies ranging from 18,000 occurrences to 2,500 ones. The results shown in the last 3 columns of Table 3 and they are indeed illuminating. The results for the low NSI group were, as expected, very low: about 4%. Those for the random set were about 15%. More interesting however are the numbers for the most frequent pairs; only 21% of these pairs were guessed, so slightly more than in the random case, and much less than with those of even moderate NSI figures. Pairs with more than 10,000 occurrences were not correctly guessed, while pairs with sometimes only a few tens of occurrences (in a 38 million-word corpus) but with high NSI, were correctly noted.

An analysis was done for some of the failures in the top 200 words of the NSI list, which showed that 12 expressions were missed consistently by the three informants. Upon verification it was found that the expressions occurred in the writings of only 3 authors out of the 90 different authors that are represented in the corpus. Clearly then we have here a case of author-specific expressions that have not spread to the general language of this literature, but were still captured by the NSI formula.

Finally the function $s_5(w)$ was computed again for all words with more than 50 occurrences and that have at least one *preceding* neighbour with local frequency greater than 10, taking into account this time *preceding* neighbours rather than following ones, and hoping therefore to retrieve frequent expressions whose *second* word is a given w. Again the list of the 300 words with the highest NSI (together with the most common preceding neighbour) was produced and it was found that most of the pairs were indeed expressions, although the proportion was markedly less than in the following-neighbour case. The same test as before was applied on this list, except that in this case the informants had to guess the *first* word of the expression, given its second one. As expected the results — given in Table 4 — were somewhat less impressive than in the first case. Although the correlation between NSI values and successful guesses was still obvious and consistent, the average percentage of success, even for the best NSI values, was only 63%, as opposed to 87% before. Such a result would certainly be understandable by anyone who would try to recite *backwards* a phrase or a small poem that he memorizes by heart. (A puzzling feature of Table 4 is the unexpected *increase* in guessings' success for *all* 3 informants when going from the 0.85–0.82 range of NSI values to the 0.82–0.81 one. We don't have as yet an explanation for that fact.)

A note should be added as an explanation to the fact that the test for preceding neighbours was based on the guessing of the *first* word of the expression, and not of the second one as before. The reason is that the second word, having a high NSE, is related to very few preceding neighbours and so there is a good chance of guessing the entire expression in spite of the psychological difficulty mentioned above. On the other hand, the first word in the retrieved sequences was usually a common word associated with a great number of different following neighbours, and by itself could not reasonably suggest any specific expression. This argument can be used to justify the suggestion that the special entry to be opened in a concordance for a retrieved expression should be related to the entry of its first word if the expression was retrieved by the following-neighbour algorithm, and to its second word in the other case.

Finally, we note that the intersection of the two produced lists of expressions was almost empty. Only about ten or so sequences were common to both lists, and these were in most cases different 2-words parts of the same 3-words expressions.

5. Conclusion

In summary, a formula is suggested that was successful in retrieving many interesting expressions in a very large corpus of Hebrew. A suitable test was derived to measure its effectiveness and was successfully applied to the retrieved list of expressions. It remains to be seen

rank	frequency	pair	ranks of the pairs' elements: first word	second word
1	18790	if not	5	1
2	17183	was not	1	24
3	16834	but if	11	5
4	12316	according to	2	323
5	11531	Maimonides	60	19
6	10751	if there is	5	18
7	9914	because if	20	5
8	9701	upon that	2	6
9	9446	that he has	55	8
10	9388	or not	18	1
11	9212	is not	1	47
12	8763	(it) he has	18	8
13	8755	not relevant	1	168
14	8483	but not	11	1
15	8404	(it) he does not have	7	8
16	7807	upon what	2	22
17	7746	not useful	1	198
18	7594	upon all	2	14
19	7588	that was not	9	24
20	7537	if it was	5	24

Table 1
Twenty most frequent pairs in a Responsa corpus of 38 million words, sorted by decreasing frequency with the frequency ranks of the pairs' elements ('first' and 'second' in the last column refer to the words in the Hebrew original pair)

what amendments or adaptations should be made to this formula to assure its successful application in smaller corpora or in other natural languages with markedly different morphological or syntactical attributes.

References

1. Choueka, Y., 'Full-text Systems and Research in the Humanities', *Computers and the Humanities*, XIV (1980), 153–169.
2. Choueka, Y., The Response Project — What, How, and Why, 1976–1981, A Status Report, Ins. for Inf. Retrieval and Computational Linguistics, Bar-Ilan University, Ramat-Gan, Israel, 1982, 55 pp.
3. Choueka, Y., Lusignan, S., 'Desambiguation by Short Contexts', to appear.
4. Martin, I., Al, B., van Sterkenburg, P., 'Text-processing and Lexicographical Information — A *State of the Art*', *ALLC Journal*, II (1981), 61–68.

NSI	f(w)	d(w)	freq. of 5 most frequent neighbours 1	2	3	4	5	# of neighbours with freq. $< M_1$	their average
1.0000	202	1	202	–	–	–	–	–	–
1.0000	192	1	192	–	–	–	–	–	–
0.9973	564	2	563	1	–	–	–	1	1
0.9899	248	2	246	2	–	–	–	1	2
0.9785	118	3	115	2	1	–	–	1	1.5
0.9611	116	2	112	4	–	–	–	1	4
0.9585	371	28	327	2	2	2	2	27	1.63
0.9142	204	2	187	17	–	–	–	–	–
0.8758	303	3	286	11	6	–	–	1	6
0.8590	110	2	95	15	–	–	–	–	–
0.7543	689	36	586	23	18	11	2	32	1.59
0.7383	1229	52	1013	53	44	31	16	47	1.53
0.7157	776	33	614	48	20	18	15	27	1.74
0.6920	2129	48	1766	146	67	30	28	41	1.41
0.6451	88	7	66	17	1	1	1	5	1
0.8311	1512	34	956	333	117	54	2	30	1.77
0.5043	146	28	51	34	10	3	2	25	2.04
0.4925	101	15	33	32	3	3	3	13	2.77
0.4556	375	56	60	54	54	46	17	48	2.15
0.4459	787	191	201	182	40	40	13	184	1.57

Table 2
Sample of a few (non-consecutive) lines of NSI values and the corresponding frequencies.

Group NSI values	1 1.00–0.95	2 0.95–0.90	3 0.90–0.87	4 0.87–0.85	5 0.85–0.83	6 0.83–0.81	7 0.63–0.60	8 0.17–0.13	9 random	10 most frequent pairs
	90	88	84	82	64	58	49	3		
	88	86	88	82	76	74	50	1		
	84	80	80	70	64	63	49	8		
Average	87	85	83	78	68	65	49	4	15	21

Table 3
Test results for 3 informants and 10 groups 50 words each, with various ranges of NSI values (following-neighbours test)

Group NSI values	1 1.0–0.92	2 0.92–0.88	3 0.88–0.85	4 0.85–0.82	5 0.82–0.81
	68	68	56	42	44
	66	54	56	50	64
	54	58	52	44	48
Average	63	59	55	45	52

Table 4
Test results for 3 informants and 5 groups of 50 words each (preceding-neighbours test)